Applying Boosting Algorithm for Improving Diagnosis of Interstitial Lung Diseases

Background and Related Work

Interstitial Lung Disease (ILD) is a group of irreversible, non-neoplastic lung pathologies that is presented as progressive scarring of lung tissue around the air sacs. The scar tissue in the lungs causes lung stiffness, which reduces the ability to breathe. While some cases of ILD are idiopathic, the general causes of ILD include autoimmune diseases, exposure to hazardous agents, and some medications[1]. If left untreated, it can lead to many life-threatening complications, such as pulmonary hypertension and respiratory failure[1].

In order to diagnosis ILDs, physicians are likely to turn to high-resolution computed tomography (HRCT) scans to identify abnormal lung tissue texture patterns. HRCT scans have enough resolution to reveal differences in the lung vasculature and to help narrow down a differential diagnosis of ILD[2]. Currently, radiologists laboriously comb through hundreds, or even thousands, of a patient's HRCT scan to look for histological abnormalities. Additionally, lung texture patterns are often difficult to distinguish, resulting in low diagnostic accuracy. Because of these reasons, there is motivation to program a computer to aid radiologists in the diagnosis of ILDs.

Currently, there are computer-assisted detection (CAD) systems that boost the confidence of physicians and serve as a "second-opinion" in their diagnoses[3]. CAD systems have been used for very specialized regions with specific imaging modalities, such as CT chest scans[4]. However, CAD technology seems to be a promising method for improving the diagnosis of various lung diseases[5]. Some state-of-the-art learning algorithms used for classifying ILDs using HRCT scans have used linear discriminant[6], Bayesian classifier[7], k-nearest neighbors[8], random forest[9], and support vector machines with linear[10], polynomial[11], and radial basis function[12] kernels. There are definite strengths and weakness of each algorithm, but since all of these methods used different datasets and features, it is difficult to compare them directly.

Having a system that could accurately diagnosis ILD could save time and money in the healthcare system. In order to improve the diagnoses, we must think about other approaches and algorithms that can be applied to this cause. Taking a look at image classification, many of the top contenders in the annual Imagenet Large Scale Visual Recognition Challenge use boosting algorithms to recognize and categorize images. This project proposes using an ensemble of weak classifiers in a boosting algorithm in an attempt to produce a high accuracy of ILD classification. The inputs to the boosting algorithm are image patches, while the outputs are predictions of different lung texture classes.

Data Source and Preprocessing

This project used the collection of images from the MedGIFT project at the University of Geneva, Switzerland[13]. It contains numerous high resolution CT scans of various cases of ILDs. In all, the collection includes three-dimensional annotated regions of lung tissues from 128 patients, who had at least one of the 13 histological diagnoses of ILD. Each region of interest was annotated with one of 17 different lung texture classes, including healthy, emphysema, ground glass, fibrosis, consolidation, micronodules, and reticulation. In addition to the images, the database also includes a spreadsheet of clinical parameters and relevant data on the patients' medical history and laboratory test results. This publicly available dataset is extremely useful for this project because of the high data quality of the healthy and pathological regions, as multiple radiologists were consulted for a consensus of a specific annotation.

In order to get the HRCT scans ready for the learning algorithm, some preprocessing was done to obtain 32x32 pixel patches. For each slice of the HRCT scan with an annotation, a bounding box was found that encompassed the whole region of interest. This bounding box was expanded, such that the dimensions of the box were multiples of 32 pixels. This box was then split into nonoverlapping 32x32 pixel images. After the patches were obtained, each patch was examined to ensure that the area of the patches overlapped with the lungs by more than 80% and the overlapped the annotated region of interest by more than 70%. An example of patch extraction can be seen in Figure 1, where the red outline corresponds to the region interest and the blue boxes are the image patches.

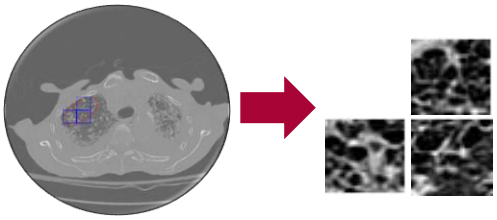


Figure 1. Example of Patch Extraction. These image patches (blue) overlap with the lung by at least 80% and with the annotated region of interest (red) by at least 70%.

Out of the 17 different texture classes, micronodules had the most image patches (see Table 1 for the breakdown of the top five texture classes). In order for all the classes to have the same number of patches, rigid transformations, such as rotations and reflections, were performed on the classes that did not have the highest patch count. For each image patch, 7 new images were created. Figure 2 provides an example of how the image transformations were performed. Rigid transformations were performed because it would not only preserve the size and shape of the patches, but also give an extra orientation for the boosting algorithm to consider in order to account for intra- and interpatient variability. Randomly chosen images were added to each class until each texture class had the same number of image patches. Classes that were unable to have sufficient patches were removed. In total, 18,705 image patches were generated after these preprocessing steps, where 80% of them were randomly taken to be the training set and the remaining 20% became the held-out test set.

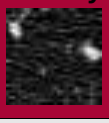
Fibrosis	Ground Glass	Healthy	Micronodules	PCP
				
1573	1777	1313	3741	592

Table 1. Counts of Texture Class. These were the top 5 texture classes counts after patch extraction. Examples of each class are presented in the table as well.

Figure 2. Example of Rigid Transformations. The original image patch was rotated and reflected in order to create additional images per texture class.

Feature Extraction and Selection

A Gabor filter was implemented in order to extract features from every image. Traditionally, Gabor filter banks are used for texture analysis and edge detection for image analysis. By observing different frequencies and orientations of images, Gabor filters have been shown to mimic how actual human visual works. Additionally, Gabor filters have been used in other medical contexts, such as pattern analysis in the directionality distribution of spinal porous trabecular bone[14]. Figure 3A is brief visual of how Gabor filters work. Ninety-six features were extracted per image patch based on the script provided by Haghighat, et al[15]. Figure 3B shows the 3-frequencies and 8-orientations Gabor filter that was used on each image patch, while Figure 3C shows the resulting images after applying that filter on an image patch.

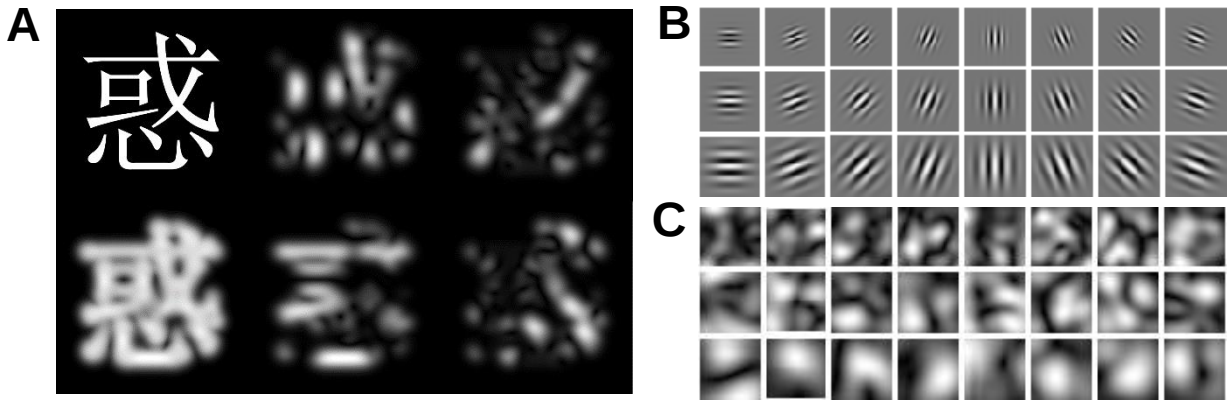


Figure 3. Representations of Gabor Filter Bank. A) The right images are filtered at 4 different orientations (0º, 45 º, 90 º, 135 º) and at 1 frequency. The left images are the original picture (top) and the superposition of the right images (bottom). B) Image of the 3-frequency, 8-orientation Gabor filter used for feature extraction. C) Example of a filtered image patch before feature extraction.

In order to choose which features were used in the learning algorithm, principal component analysis (PCA) was implemented[16]. PCA uses orthogonal transformation to convert the set of features into linearly uncorrelated variables in order to select the features that explain nearly all of the variance in the dataset. In order to maximize the variance of each feature, the values were scaled to a unit-length u to maximize the following equation:

$$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)^T}u)^2 = \frac{1}{m}\sum_{i=1}^{m}u^T x^{(i)}x^{(i)^T}u$$
$$= u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)^T}\right)u.$$

After running PCA, a cutoff was placed at the features that explained more than 0.1% of the variance. This cutoff boundary was based on the drop-off in proportion of variance explained by the feature, as shown in Figure 4. In the end, of the 96 features, 72 were selected to be inserted in the boosting algorithm.

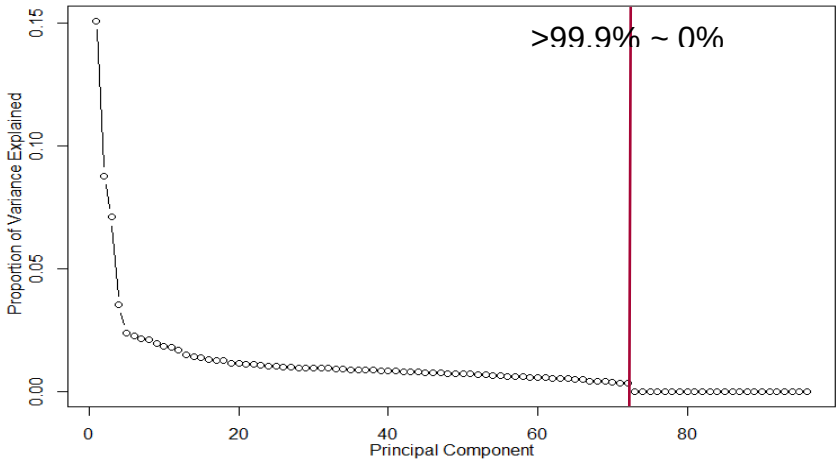


Figure 4. Graph of Principal Components. The first 72 features explain more than 0.1% of the variance, which sums up to more than 99.9% of the variance. There was a drop off of variance explained after the 72th feature.

Methods

This project utilized the gradient boosted model (gbm) package in R[17], which implements extensions to Freund and Schapire's AdaBoost algorithm[18] and Friedman's gradient boosting machine[19,20]. The basics of boosting models are that they use an ensemble of weak predictive models, most notably decision trees. As with many supervised learning algorithms, given a training set $\{(x_1, x_2), \ldots, (x_n, y_n)\}$, there is a desire to minimize the expected value of some specific loss function $L(y, F(x))$:

$$\hat{F} = \arg\min_{F} \mathbb{E}_{x,y}[L(y, F(x))]$$

Gradient boosting approximates $\hat{F}$ as a weighted sum of weak learners of function $h_i(x)$:

$$F(x) = \sum_{i=1}^{M}\gamma_i h_i(x) + \text{const}$$

When using trees to fit this model, the entire input is partitioned into disjoint regions $R_{1m}, \ldots, R_{jm}$. Thus, these functions can be calculated as $h_m(x)$:

$$h_m(x) = \sum_{j=1}^{J} b_{jm} I(x \in R_{jm}),$$

where b_{jm} is a value predicted in R_{jm}.

Friedman modified this algorithm by proposing the use of a different parameter γ_{jm} for each region, thereby replacing the b_{jm} parameter initially proposed. The final gradient boosting model uses this update rule:

$$F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J}\gamma_{jm}h_m(x)I(x \in R_{jm}), \quad \gamma_{jm} = \arg\min_{\gamma}\sum_{x_i \in R_{jm}} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)).$$

Results [Different from what was submitted on the poster]

By using the gbm package in R, different parameters were able to be tuned, in order to find the best model for this project's classification problem. The number of trees ranged from 10 to 400 trees, where each data point in Figure 5 was an addition of 10 more trees from the previous iteration. Another parameter tuned by the model was the interaction depth

at each stage of the decision tree. This parameter controls the number of splits at each tree, starting from the single base node. Hastie et al proposed the range of 4 to 8 splits per tree to be a comfortable range that allows for interaction between variables in the model[21]. Increasing the number of trees and the number of splits increases the complexity of the model and may lead to overfitting; however, some regularization parameters were also tuned to accommodate for any overfitting issues the model may have generated.

One of the regularization parameters is shrinkage, which affects the impact of additional fitted trees by modifying the update rule:

$$F_m(x) = F_{m-1}(x) + \nu \cdot \gamma_m h_m(x), \quad 0 < \nu \leq 1$$

In other words, this parameter specifies the learning rate of how the algorithm adapts to seeing new data and penalizes the importance of each subsequent iteration. On the flipside, lowering the learning rate may increase the number of iterations the model takes to converge. Hastie, et al. also found that small learning rates, such as $\nu < 0.1$, have better generalization ability over the models without any shrinkage factor[21]. Therefore, this project looked at shrinkage factors between 0.03 and 0.09. The second regularization term being tuned was the minimum number of observations in the trees' terminal nodes. The splits with fewer than the specified number will be ignored. By varying this parameter, the variance of the model may decrease, leading to better accuracy. This project looked at the difference between having 3 and 4 terminal nodes. Lastly, a 3-fold cross-validation was implemented in order to ensure better accuracy when training the model at each given parameter. See Figure 5 for the difference in accuracy as tuned by varying different parameters.

The final model used 400 trees, 4 terminal nodes, 8 interaction depth (splits) per tree and a shrinkage factor of 0.09. This model was used to predict the label on a held-out test set of 3,741 image patches. The gradient boosting algorithm produced probabilities of each test image having each of the 5 labels. From that result, the highest probability was taken to be the predicted label of the test image. Ultimately, the model achieved 67% accuracy, where the confusion matrix is presented in Table 2. Table 3 contains additionally information regarding different metrics of comparison between the different texture classes, such as sensitivity, specificity, and detection rate.

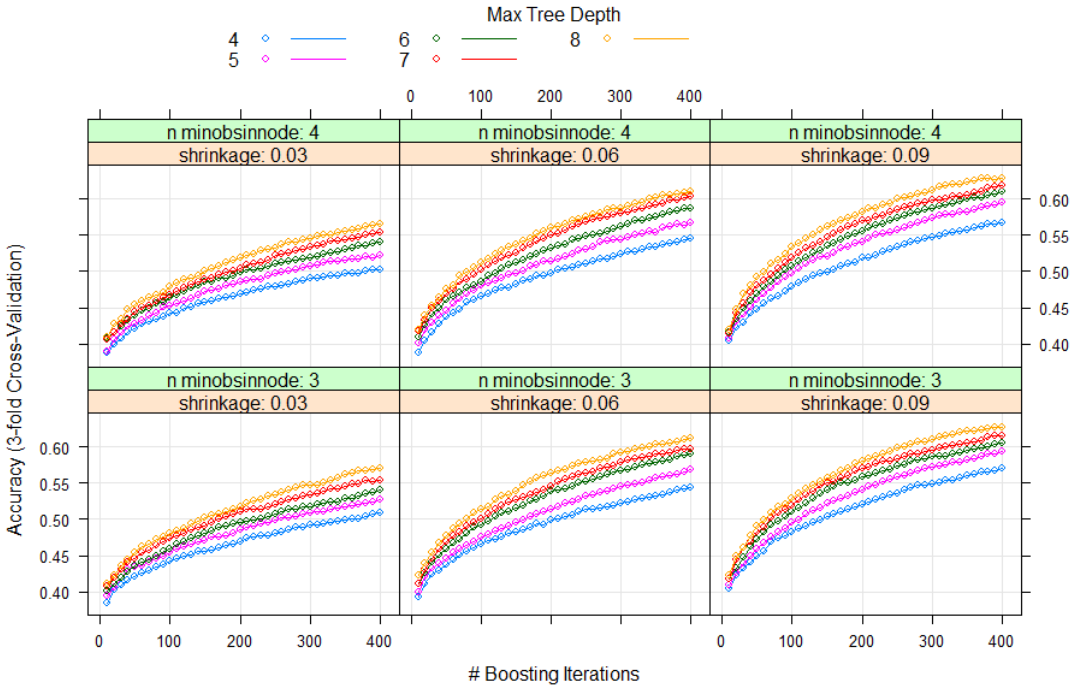


Figure 5. Parameter Tuning of the Gradient Boosted Model. The boosting modeling was run for a range of 10-400 trees, with 10 tree intervals in between data points. The interaction depth ranged from 4-8 splits per tree, while the terminal nodes were either 3 or 4 per leaf of the decision tree. Shrinkage ranged from 0.03 to 0.09.

The results seem to indicate that the best classification occurred when predicting micronodules. This lung texture was the class that did not have any data augmentation done on it during the preprocessing step, which may indicate that the preprocessing step might have been a mistake. Rerunning the boosting algorithm using 360 trees, 8 interaction depths, 4 terminal nodes, and 0.09 shrinkage factor on the dataset without image augmentation yielded an accuracy of 0.8438. The training set had 7197 images, while the test set had 1799 images. Table 4 is the confusion matrix for this boosting algorithm, while Table 5 shows the same statistical summary as before. Figure 6 shows qualitative examples of misclassification from the boosting algorithm. This result comes as a surprise at the end of the project; therefore, future

research should be done to see the effects of data augmentation. On the other hand, an additional dataset of images could be acquired to increase the robustness of the data.

		Reference				
		Fibrosis	Ground Glass	Healthy	Micro-nodules	PCP
Predicted	Fibrosis	441	70	81	45	52
	Ground Glass	87	421	91	28	80
	Healthy	78	89	443	62	27
	Micro-nodules	58	33	96	605	10
	PCP	80	134	29	6	595

Table 2. Confusion Matrix of the Test Set. The overall accuracy of the model on the test set was 67%.

	Fibrosis	Ground Glass	Healthy	Micro-nodules	PCP
Sensitivity	0.5927	0.5636	0.5986	0.8110	0.7788
Specificity	0.9173	0.9045	0.9147	0.9342	0.9164
Positive Predictive Value	0.6401	0.5955	0.6388	0.7544	0.7050
Negative Predictive Value	0.9007	0.8926	0.9024	0.9520	0.9417
Detection Rate	0.1179	0.1125	0.1184	0.2144	0.2256

Table 3. Statistical Summary of the Texture Classes of the Test Set. This indicates Ground Glass is the hardest to detect, while Micronodules are the easiest to detect using gradient boosted models.

		Reference				
		Fibrosis	Ground Glass	Healthy	Micro-nodules	PCP
Predicted	Fibrosis	283	22	15	3	9
	Ground Glass	19	228	16	8	10
	Healthy	11	0	209	16	2
	Micro-nodules	38	12	81	698	5
	PCP	9	5	0	0	100

Table 4. Confusion Matrix of the Test Set Without Data Augmentation. The overall accuracy of the model on the test set was 84.38%.

	Fibrosis	Ground Glass	Healthy	Micro-nodules	PCP
Sensitivity	0.7861	0.8539	0.6511	0.9628	0.7937
Specificity	0.9659	0.9654	0.9804	0.8734	0.9916
Positive Predictive Value	0.8524	0.8114	0.8782	0.8369	0.8772
Negative Predictive Value	0.9475	0.9743	0.9283	0.9720	0.9846
Detection Rate	0.1573	0.1267	0.1162	0.3880	0.0556

Table 5. Statistical Summary of the Texture Classes of the Test Set Without Data Augmentation. This indicates Healthy is the hardest to detect, while Micronodules are the easiest to detect using gradient boosted models.

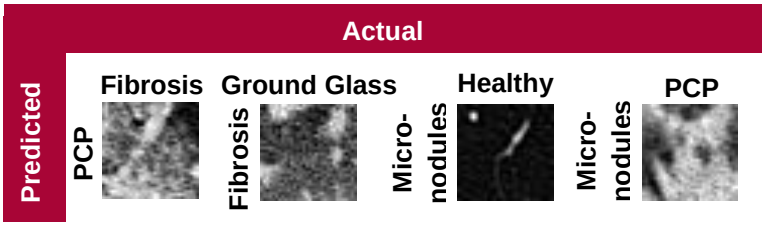


Figure 6. Qualitative Results of Misclassifications. Some of the misclassifications that the second boosting algorithm made. In the future, features regarding pixel intensity should be added to increase the accuracy of classification.

Future Work

In all, this project explored the use of a boosting algorithm in the application of diagnosing interstitial lung disease. This project was able to achieve 67%/84.38% accuracy, which may be an unacceptable rate in the medical field. Physicians and radiologist would need a very accurate predictor assisting them in practice. This result was unexpected, but at least performed better than randomly predictions, which would have had an accuracy of 20%. The features extracted from a Gabor filter bank might not have been enough for the algorithm to achieve accuracy.

The future work for this project would be to include a more robust feature selection step. Sequential feature search was attempted in the feature selection set, but that algorithm produced negligible results. Additionally, other features can also be extracted from the image patches to complement the Gabor filter, such as histogram of oriented gradients. Another improvement to this project would be to use a more powerful machine learning algorithm. Recently, there has been efforts in using convoluted neural networks to improve the diagnosis of ILD[22]. Finding a way to combine a neural network with ensemble learning would likely produce great results.

References

1. “Interstitial lung disease,” *Mayo Clinic*, Jun. 2015. [Online]. Accessed: Nov. 11 2016.
2. B. Elicker, C. A. de C. Pereira, R. Webb, and K. O. Leslie, "High-resolution computed tomography patterns of diffuse interstitial lung disease with clinical and pathological correlation," *Jornal Brasileiro de Pneumologia*, vol. 34, no. 9, pp. 715–744, Sep. 2008.
3. K. Doi, "Computer-aided diagnosis in medical imaging: Historical review, current status and future potential," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 198–211, Jun. 2007.
4. R. M. Summers, "Road maps for advancement of Radiologic computer-aided detection in the 21st century," *Radiology*, vol. 229, no. 1, pp. 11–13, Oct. 2003.
5. U. Bağcı, M. Bray, J. Caban, J. Yao, and D. J. Mollura, "Corrigendum to ‘Computer-assisted detection of infectious lung diseases: A review’ [*Comput. Med. Imag. Graph*. 36 (2012) 72–84]," *Computerized Medical Imaging and Graphics*, vol. 36, no. 2, p. 169, Mar. 2012.
6. I. C. Sluimer, P. F. van Waes, M. A. Viergever, and B. van Ginneken, "Computer-aided diagnosis in high resolution CT of the lungs," *Medical Physics*, vol. 30, no. 12, pp. 3081–3090, Nov. 2003.
7. Y. Xu, E. J. R. van Beek, Y. Hwanjo, J. Guo, G. McLennan, and E. A. Hoffman, "Computer-aided classification of interstitial lung diseases via MDCT: 3D Adaptive multiple feature method (3D AMFM)," *Academic Radiology*, vol. 13, no. 8, pp. 969–978, Aug. 2006.
8. V. Sajwan, "Content based image retrieval using combined features (color and texture)," *International Journal of Engineering Research*, vol. 3, no. 4, pp. 271–273, Apr. 2014.
9. M. Anthimopoulos, S. Christodoulidis, A. Christe, S. Mougiakakou, "Classification of interstitial lung disease patterns using local DCT features and random forest", *Proc. 36th Annual International Conference IEEE Engineering in Medicine and Biology Society*, pp. 6040-6043, 2014.
10. Q. Li, W. Cai, D. D. Feng, "Lung image patch classification with automatic feature learning", *Proc. Annual International Conference IEEE Engineering in Medicine and Biology Society*, pp. 6079-6082, 2013.
11. Y. Song, W. Cai, Y. Zhou, and D. D. Feng, "Feature-based image patch approximation for lung tissue classification," *IEEE Transactions on Medical Imaging*, vol. 32, no. 4, pp. 797–808, Apr. 2013.
12. M. J. Gangeh, "A texton-based approach for the classification of lung parenchyma in CT images", *Proc. MICCAI*, pp. 595-602, 2010.
13. A. Depeursinge, A. Vargas, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Building a reference multimedia database for interstitial lung diseases," *Computerized Medical Imaging and Graphics*, vol. 36, no. 3, pp. 227–238, Apr. 2012.
14. C. M. Gdyczynski, A. Manbachi, S. Hashemi, B. Lashkari, and R. S. C. Cobbold, "On estimating the directionality distribution in pedicle trabecular bone from micro-CT images," *Physiological Measurement*, vol. 35, no. 12, pp. 2415–2428, Nov. 2014.
15. M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7905–7916, Nov. 2015.
16. "Practical guide to principal component analysis (PCA) in R & python," *Analytics Vidhya*, 2016. [Online]. Accessed: Dec. 9, 2016.
17. G. Ridgeway, “Generalized Boosted Regression Models,” *CRAN*, 2015.
18. Y. Freund and R. E. Schapire, "A Decision-Theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
19. J. H. Friedman, " Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
20. J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
21. T. Hastie, J. Friedman, and R. Tibshirani, “10. Boosting and Additive Trees,” *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed. New York, NY: Springer-Verlag New York, pp. 337–384, 2009.
22. M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep Convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1207–1216, May 2016.